Motivating MEMS 1140

Week 1

MEMS 1140—Introduction to Programming in Mechanical Engineering

Learning Objectives (L.O.)

At the end of this lecture, you should understand/be able to:

- ❑ The engineering motivation for learning programming;
- ❑ The mission of MEMS 1140;
- ❑ The structure of MEMS 1140;
- ❑ Applications of coding in Mechanical Engineering at Pitt;
- ❑ The objectives of MEMS 1141 & 1142.

Table of Contents (ToC)

1 – The Burning Question

I am a mechanical engineering student.

Why should I learn how to code?

1 – National Academy of Engineering

A group of experts that guides the field of engineering.

Let's consider and reflect on two excerpts from their report: "The Engineer of 2020: Visions of Engineering in the New Century"

1 – Motivation from the NAE

Regarding the engineering profession:

1 – Motivation from the NAE

⇨ L.O.1
❏ L.O.2
❏ L.O.3
❏ L.O.4
❏ L.O.5

Regarding the engineering profession:

> *we must ask if it serves [us] well to permit [engineering] to lag technology and society, especially as technological change occurs at a faster and faster pace. Rather, should the engineering profession anticipate needed advances and prepare for a future where it will provide more benefit to human-kind?* [1]

[1] *The Engineer of 2020: Visions of Engineering in the New Century*, National Academy of Engineering, US, 2004.

1 – Motivation from the NAE

Regarding the engineering profession:

> *we must ask if it serves [us] well to permit [engineering] to lag technology and society, especially as technological change occurs at a faster and faster pace. **Rather, should the engineering profession anticipate needed advances and prepare for a future where it will provide more benefit to human-kind?*** [1]

[1] *The Engineer of 2020: Visions of Engineering in the New Century*, National Academy of Engineering, US, 2004.

1 – Motivation from the NAE

Regarding tools in engineering:

> *The most important new tool, however, has been the computer. . . It has allowed us to study very large numbers of examples, millions instead of dozens.*[1]

[1] *The Engineer of 2020: Visions of Engineering in the New Century*, National Academy of Engineering, US, 2004.

1 – Reflection

Question: Is it not our responsibility to adapt ahead of the curve of technology?

Assertion: The computer is the *most important* tool for a modern engineer.

1 – Takeaway

Engineers ***must*** become familiar with computing to remain competent in the modern world.

2 – The Mission

The goal of MEMS 1140 is for students to develop proficiency with computer programming such that they are able to:

- Develop and implement solution algorithms for engineering problems;
- Build basic programming literacy;
- Continue to MEMS 1141 (non/linear regression) & 1142 (Machine Learning).

3 – The Method

Example-based: programming is introduced through examples within the Mechanical Engineering discipline.

Relevance: all material will be motivated by examples you have already seen, or will see in ENGR 0135 (Statics 1).

Code-centric: You will learn ***how to code*** to solve problems traditionally done with pen and paper.

3 – The Structure

✓ L.O.1
✓ L.O.2
➩ L.O.3
❏ L.O.4
❏ L.O.5

Out of Class: (approx. 30 minutes)

- Lecture videos–view *before* class
- Conceptual checkpoint questions–complete *while watching* the videos

In Class: (50 minutes)

- Programming practicum–apply new knowledge to solve exciting problems!

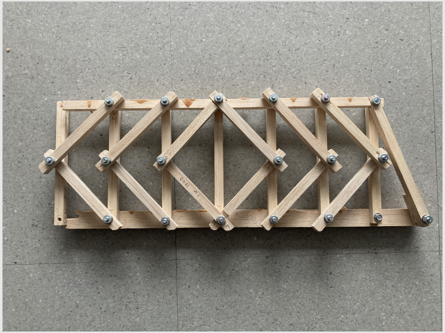
4 – Real Projects

The following are examples of projects that utilize coding in Mechanical Engineering classes.

These projects are far too involved to be reasonably done with pen and paper calculations.

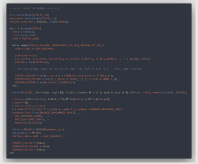
Coding enables more robust and complex analyses.

4.1— ENGR 0135 Bridge



Goal: Design, Optimize, and Construct a balsa wood bridge.

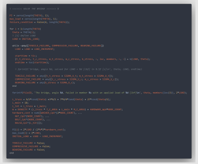
4.1— ENGR 0135 Bridge

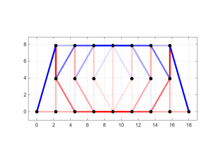


Code instructs the computer to solve the bridge automatically.

4.1— ENGR 0135 Bridge

✓ L.O.1
✓ L.O.2
✓ L.O.3
➩ L.O.4
❏ L.O.5

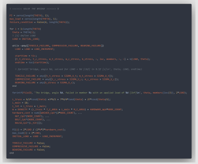


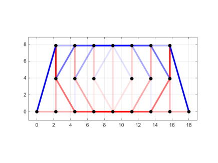


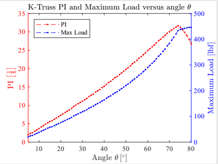
This enables very useful visualizations. The graph here highlights the load on each member in the truss.

4.1— ENGR 0135 Bridge

✓ L.O.1
✓ L.O.2
✓ L.O.3
➩ L.O.4
❑ L.O.5



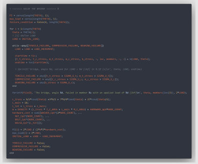


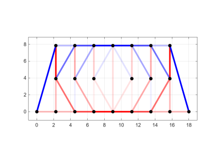


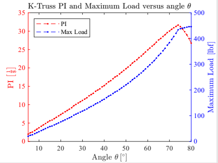
Optimization is very simple by sweeping over the entire design space. The peak of PI represents the “best” bridge.

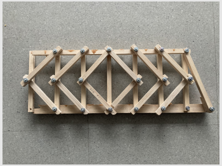
4.1— ENGR 0135 Bridge

- ✓ L.O.1
- ✓ L.O.2
- ✓ L.O.3
- ➩ L.O.4
- ❏ L.O.5



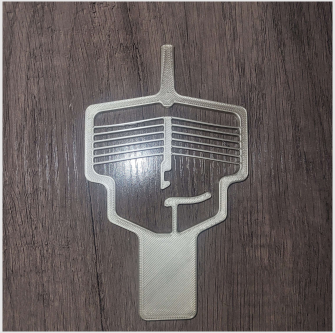






Ultimately, the real bridge is informed by a robust optimization study.

4.2— MEMS 1120 Crossbow



Goal: Optimize the design for a 3D printed toothpick crossbow.

4.2— MEMS 1120 Crossbow

- ✓ L.O.1
- ✓ L.O.2
- ✓ L.O.3
- ➪ L.O.4
- ❑ L.O.5

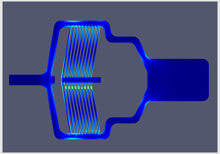
```python
sp.start_session()
igs_files = hf.get_igs_files("FINAL_SIM_FILES")

for igs_file in igs_files:
    print(igs_file)
    [rib_thick, num_rib] = hf.parse_filename(igs_file)
    if  1: #rib_thick == 1:
        sp.import_geom(hf.create_full_path(igs_file))
        sp.element_setup(5.5, EX = 1700)
        sp.mesh_model(3, 0, 0)
        sp.apply_bcs()
        sp.gen_results()
        print(sp.get_num_nodes())
        #sp.save_eplot(igs_file)
        #sp.save_eqvstress_plot(igs_file)

        reaction_forces.loc[num_rib, rib_thick] = sp.get_reactionforce()
        max_stress.loc[num_rib, rib_thick] = sp.get_max_eqvstress()
        num_elems.loc[num_rib, rib_thick] = sp.get_num_elems()
        num_nodes.loc[num_rib, rib_thick] = float(sp.get_num_nodes())

sp.mapdl.exit()
```

Code automates SolidWorks to generate many variations of the geometry.

4.2— MEMS 1120 Crossbow

- ✓ L.O.1
- ✓ L.O.2
- ✓ L.O.3
- ➩ L.O.4
- ❏ L.O.5

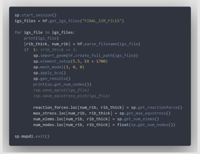
```
sp.start_session()
igs_files = hf.get_igs_files("FINAL_SIM_FILES")

for igs_file in igs_files:
    print(igs_file)
    [rib_thick, num_rib] = hf.parse_filename(igs_file)
    if  1: #rib_thick == 1:
        sp.import_geom(hf.create_full_path(igs_file))
        sp.element_setup(5.5, EX = 1700)
        sp.mesh_model(3, 0, 0)
        sp.apply_bcs()
        sp.gen_results()
        print(sp.get_num_nodes())
        #sp.save_eplot(igs_file)
        #sp.save_eqvstress_plot(igs_file)

        reaction_forces.loc[num_rib, rib_thick] = sp.get_reactionforce()
        max_stress.loc[num_rib, rib_thick] = sp.get_max_eqvstress()
        num_elems.loc[num_rib, rib_thick] = sp.get_num_elems()
        num_nodes.loc[num_rib, rib_thick] = float(sp.get_num_nodes())

sp.mapdl.exit()
```



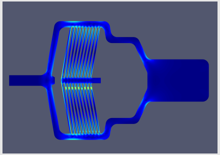
Each variation is automatically simulated in Ansys to get a stress profile.

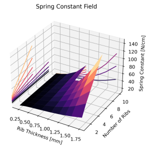
4.2— MEMS 1120 Crossbow

- ✓ L.O.1
- ✓ L.O.2
- ✓ L.O.3
- ➩ L.O.4
- ❑ L.O.5



```python
sp.start_session()
igs_files = hf.get_igs_files("FINAL_SIM_FILES")

for igs_file in igs_files:
    print(igs_file)
    [rib_thick, num_rib] = hf.parse_filename(igs_file)
    if  1: #rib_thick == 1:
        sp.import_geom(hf.create_full_path(igs_file))
        sp.element_setup(5.5, EX = 1700)
        sp.mesh_model(3, 0, 0)
        sp.apply_bcs()
        sp.gen_results()
        print(sp.get_num_nodes())
        #sp.save_eplot(igs_file)
        #sp.save_eqvstress_plot(igs_file)

        reaction_forces.loc[num_rib, rib_thick] = sp.get_reactionforce()
        max_stress.loc[num_rib, rib_thick] = sp.get_max_eqvstress()
        num_elems.loc[num_rib, rib_thick] = sp.get_num_elems()
        num_nodes.loc[num_rib, rib_thick] = float(sp.get_num_nodes())

sp.mapdl.exit()
```

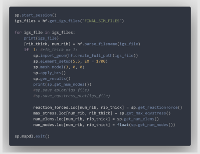




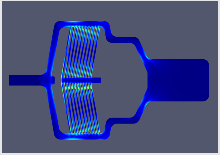
Optimization is very simple by studying the spring behavior for each variation. The “best” crossbow stores the most elastic energy.

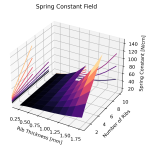
4.2— MEMS 1120 Crossbow

✓ L.O.1
✓ L.O.2
✓ L.O.3
➩ L.O.4
❑ L.O.5



```
sp.start_session()
igs_files = hf.get_igs_files("FINAL_SIM_FILES")

for igs_file in igs_files:
    print(igs_file)
    [rib_thick, num_rib] = hf.parse_filename(igs_file)
    if  1: #rib_thick == 1:
        sp.import_geom(hf.create_full_path(igs_file))
        sp.element_setup(5.5, EX = 1700)
        sp.mesh_model(3, 0, 0)
        sp.apply_bcs()
        sp.gen_results()
        print(sp.get_num_nodes())
        #sp.save_eplot(igs_file)
        #sp.save_eqvstress_plot(igs_file)

        reaction_forces.loc[num_rib, rib_thick] = sp.get_reactionforce()
        max_stress.loc[num_rib, rib_thick] = sp.get_max_eqvstress()
        num_elems.loc[num_rib, rib_thick] = sp.get_num_elems()
        num_nodes.loc[num_rib, rib_thick] = float(sp.get_num_nodes())

sp.mapdl.exit()
```





Ultimately, the real crossbow is informed by a robust optimization study.

4.2— Other MEMS Courses

✓ L.O.1
✓ L.O.2
✓ L.O.3
➪ L.O.4
❏ L.O.5

Other mechanical engineering courses that utilize coding:

- Statics 1 & 2;
- Thermodynamics;
- Circuits;
- Fluid Mechanics;
- Dynamic Systems;
- Measurements 1 & 2;
- Mechanical Design 2;
- Automatic Controls;
- Mechatronics;
- Applied Simulation in Engineering Design;

4.2— Other MEMS Courses

Other mechanical engineering courses that utilize coding:

- Statics 1* & 2;
- Thermodynamics*;
- Circuits*;
- Fluid Mechanics*;
- Dynamic Systems*;
- Measurements 1* & 2*;
- Mechanical Design 2;
- Automatic Controls*;
- Mechatronics;
- Applied Simulation in Engineering Design;

**Require the use of MATLAB*

4.2— Other MEMS Courses

✓ L.O.1
✓ L.O.2
✓ L.O.3
⇨ L.O.4
❏ L.O.5

Other mechanical engineering courses that utilize coding:

- Statics 1*† & 2†;
- Thermodynamics*†;
- Circuits*†;
- Fluid Mechanics*†;
- Dynamic Systems*†;
- Measurements 1*† & 2*†;
- Mechanical Design 2†;
- Automatic Controls*;
- Mechatronics;
- Applied Simulation in Engineering Design;

**Require the use of MATLAB*

† Required in the curriculum

5 – Beyond MEMS 1140

MEMS 1141:

- Introduces the foundations of linear and nonlinear regression analyses in Mechanical Engineering.

MEMS 1142:

- Applies linear and nonlinear regression analyses to build Machine Learning models for Dynamics, Fluid Mechanics, Heat Transfer, etc.

6 – Summary

✓ L.O.1
✓ L.O.2
✓ L.O.3
✓ L.O.4
✓ L.O.5

This lecture covered:

- ✓ The engineering motivation for learning programming

  The NAE emphasizes that learning how to code is essential for engineers to remain ahead of technological development.

- ✓ The objectives of MEMS 1140

  This course will build competency in programming for Mechanical Engineering to address the expectations set out by the NAE.

6 – Summary

✓ L.O.1
✓ L.O.2
✓ L.O.3
✓ L.O.4
✓ L.O.5

✓ The structure of MEMS 1140

This course follows a flipped class format; lecture videos teach core concepts outside of class, and class time is dedicated to applied practice.

✓ Applications of coding in Mechanical Engineering at Pitt

Programming enables more robust projects, as seen with ENGR 0135 and MEMS 1120, plus many more.

6 – Summary

✓ L.O.1
✓ L.O.2
✓ L.O.3
✓ L.O.4
✓ L.O.5

✓ The objectives of MEMS 1141 & 1142

MEMS 1141 covers linear and nonlinear regression;

MEMS 1142 covers Machine Learning in Mechanical Engineering.